NetApp®

Go further, faster®

NFSv4.2 Secure Inter-server Server Side Copy Status

William A. (Andy) Adamson

andros@netapp.com

IETF 89, London

History

- IETF87: No progress on draft-ietf-nfsv4-rpcsec-gssv3
- Discussion of draft-20, which removed the GSSv3 requirement, exposed several issues with non-GSSv3 secure inter-server server side copy
  - Several choices but no clear solution from list discussions
- IETF88: Decipher and present choices from list to WG
- IETF89: New draft-ietf-nfsv4-rpcsec-gssv3 chosen for NFSv4.2 Inter SSC security and transport of server security labels.

Secure Inter-server SSC Goals Review

1. Source server properly authenticates the destination server
2. Destination server READ is associated with the copy and is handled in a special manner by the source (see READ stateid issue slide)
3. Destination server is granted the privilege to act on behalf of the user-principal to READ.
4. Limit the ability of the destination server to act as the user-principal (e.g. a single copy)

READ Stateid Issue: use of ca_src_stateid

- COPY ca_src_stateid is from the client OPEN verified against the client clientid (NFSv4.1)
- Destination (acting as a client) to perform ‘normal’ READs from the source
  - No OPEN from the destination server to avoid (share) locking issues
  - Like to READ with ca_src_stateid and the COPY SAVE_FH
- Source needs to know the READ stateid is special
  - So as not to verify it against the destination server clientid

NFSv4.2 SSC use of RPCSEC_GSSv3

- Used for ***inter-server*** server-side copy
- A generated ‘shared’ secret plus user-principal info is distributed to and between the source and destination via RPCSEC_GSS3_CREATE calls
  - NFSv4.2 defines several RPCSEC_GSS version 3 structured privileges
  - Compound authentication is also required to authorize the destination server to act on behalf of the user principal

NFSv4.2 SSC use of RPCSEC_GSSv3

- Information is distributed via a series of GSSv3 structured privilege assertions sent with Privacy enabled
- The ‘shared secret is distributed first to the source, then to the destination, and finally is presented by the destination to the source as an identifier for the particular copy.
- User principal information required for compound authentication is passed from the client to the destination and then from the destination to the source.

RPCSEC_GSS_CREATE Calls

- A copy_from_auth privilege GSSv3 context is used to send the COPY_NOTIFY from the client to the source
- A copy_to_auth privilege GSSv3 context is used to send the COPY from the client to the destination
- A copy_confirm_auth privilege plus compound_auth GSSv3 context is used for the destination READs from the source
  - The copy_from_auth and copy_to_auth exist to establish the copy_confirm_auth privilege.

NFSv4.2 Inter SSC Step 1

- The user principal establishes an RPCSEC_GSSv3 context with the source server (princ-src context)
- The user principal OPENs the file to be copied on the source server using the princ-src context.
- A COPY_NOTIFY is sent to the source server using a copy_from_auth structured privilege GSSv3 handle.

copy_from_auth

- A user principal is authorizing a source principal (nfs/<source>) to allow a destination principal (<nfs/destination>) to setup the copy_confirm_auth privilege.
- Established on the source server before the COPY_NOTIFY operation is sent to the source server, and the resultant context is used to secure the COPY_NOTIFY operation.

copy_from_auth: RPCSEC_GSS_CREATE

```
struct copy_from_auth_priv {
        secret4          cfap_shared_secret;
        netloc4          cfap_destination;
        /* the NFSv4 user name that the user principal maps to */
        utf8str_mixed    cfap_username;
  };
```

NFSv4.2 Inter SSC Step 2

- The user principal establishes an RPCSEC_GSSv3 context with the destination server (princ-dst context)
- The user principal OPENs the file to be copied to on the destination server using the princ-dst context.
- A COPY is sent to the source server using a copy_to_auth structured privilege GSSv3 handle.
  - The copy_to_auth privilege grants the destination server the ability to setup a **compound authentication** assertion with the source server.

NFSv4.2 SSC & Compound Authentication

- For compound authentication to work, the user principal's context handle, a nonce, and a MIC of the nonce using the user principals context is sent in an RPCSEC_GSS_CREATE rgss3_gss_binding payload.
  - The user principal handle is the "inner" handle
  - The client machine principal is the parent handle.
- The target verifies the inner handle by locating the inner handle context, and calling GSS_VerifyMIC on the nonce

NFSv4.2 SSC & Compound Authentication

- For inter SSC, the user principal has no context established with the source server on the destination
- As noted in step 1, the client and the source share a princ-src context which is used for the OPEN of the source file to be copied.
- The copy_to_auth privilege creates the compound authentication payload that the destination server will use to establish a compound authorization with the source
  - E.G. all of the ‘inner handle’ data

copy_to_auth

- A user principal is authorizing a destination principal (nfs/<destination>)to setup the copy_confirm_auth privilege with a source principal (nfs/<source>).
- Established on the destination server before the COPY operation is sent to the destination server, and the resultant context is used to secure the COPY operation.

copy_to_auth: RPCSEC_GSS_CREATE

```
struct copy_to_auth_priv {
        /* equal to cfap_shared_secret */
        secret4          ctap_shared_secret;
        netloc4          ctap_source;
        /* the NFSv4 user name the user principal maps to */
        utf8str_mixed    ctap_username;
        opaque           ctap_handle;  ←princ-src context handle
        /* A nounce and a mic of the nounce using ctap_handle */
        opaque           ctap_nounce;
        opaque           ctap_nounce_mic;
 };
```

copy_confirm_auth

- A destination principal (nfs/<destination>) is confirming with the source principal (nfs/<source>) that it is authorized to copy data from the source on behalf of the user principal.
- Established on the destination server before the file is copied from the source to the destination, and the resultant context is used to secure READ operations from the source to the destination.
- Note that the resultant GSSv3 handle MUST be destroyed by the destination if the copy_to_auth privilege handle is destroyed.

copy_to_auth: RPCSEC_GSS_CREATE

```
struct copy_confirm_auth_priv {
        /* equal to GSS_GetMIC() of cfap_shared_secret */
        opaque              ccap_shared_secret_mic<>;
        /* the NFSv4 user name that the user principal maps to */
        utf8str_mixed       ccap_username;
};
struct rgss3_gss_binding {
        opaque          rgb_handle<>; /* inner handle */
        opaque          rgb_nonce<>;
        opaque          rgb_nounc_mic<>;
};
```

RPCSEC_GSS3

- Authenticates the destination server
  - YES, via the shared secret distributed via GSS3
- Destination READ special handling at source
  - YES, using the copy_confirm_auth GSS3 handle for READs
- Act on behalf of the user-principal
  - YES, via the use of compound authentication for the copy_confirm_auth GSS3 context handle creation
- Limit the destination server
  - YES, client destroys the copy_from_auth and copy_to_auth GSS3 context handles

NetApp®

Thank you